On the Complexity of the Minimum Independent Set Partition Problem

T-H. Hubert Chan[1], Charalampos Papamanthou[2], and Zhichao Zhao[1]

[1] Department of Computer Science
the University of Hong Kong
{hubert,zczhao}@cs.hku.hk
[2] Department of Electrical and Computer Engineering and UMIACS
University of Maryland
cpap@umd.edu

Abstract. We consider the Minimum Independent Set Partition Problem (MISP) and its dual (MISPDual). The input is a multi-set of N vectors from $\{0,1\}^n$, where $U := \{1,\ldots,n\}$ is the index set. In MISP, a threshold k is given and the goal is to partition U into a minimum number of subsets such that the projected vectors on each subset of indices has multiplicity at least k, where the multiplicity is the number of times a vector repeats in the (projected) multi-set. In MISPDual, a target number χ is given instead of k, and the goal is to partition U into χ subsets to maximize k such that each projected vector appears at least k times.

The problem is inspired from applications in private voting verification. Each of the N vectors corresponds to a voter's preference for n contests. The n contests are partitioned into χ subsets such that each voter receives a verifiable tracking number for each subset. For each subset of contests, each voter's tracking number together with the votes for that subset is released in some public bulletin, which can be verified by each voter. The multiplicity k of the vectors' projection onto each subset of indices ensures that the bulletin for each subset of contests satisfies the standard privacy notion of k-anonymity.

In this paper, we show strong inapproximability results for both problems. For MISP, we show the problem is hard to approximate to within a factor of $n^{1-\epsilon}$. For MISPDual, we show the problem is hard to approximate to within a factor of $N^{1-\epsilon}$. Here, ϵ can be any small constant. Note that factors n and N approximation are trivial for MISP and MISPDual respectively. Hence, our results imply that any polynomial-time algorithm can almost do no better than the trivial one.

1 Introduction

We study the Minimum Independent Set Partition problem (MISP) and its dual problem (MISPDual). This problem was raised by Wagner on cstheory.stackexchange [12] in the context of data privacy [6]. We first describe the problem and an application scenario.

In MISP, a multi-set Y of N vectors in $\{0,1\}^n$ is given together with a multiplicity threshold k. Our goal is to partition the indices $[n]$ into minimum number χ of subsets such that the projection of Y on each subset has multiplicity at least k.

The dual problem MISPDual is also of interest, in which a multi-set Y of vectors is also given. However, the target number χ of parts is given, and the goal is to return a χ-partition of the indices $[n]$ such that the minimum multiplicity k of the projected vectors is maximized.

Application Scenario. The problem is motivated by privacy in voting verification. We have N voters, each of whom is voting for n contests (with $\{0,1\}$ voting). To verify that all votes have been counted, each voter gets assigned a verifiable tracking number during voting. Then, there is a public bulletin board where all pairs of tracking numbers and votes are posted (where names of voters are withheld) such that each voter can verify that his votes are correct using his own tracking number. This could provide verifiability, but it is well-known in the privacy community that simply replacing a user's name with a random id cannot achieve privacy [6], since a voter might be uniquely identified by the way he votes in the n contests.

An expensive solution would be for each voter to get a separate tracking number for each contest, but this would increase the space complexity to store n numbers for each voter. Observe that if k is the minimum of the number of minority votes over all n contests, this expensive solution achieves the standard notion of k-anonymity [11].

To obtain a tradeoff between the space complexity of each voter and the anonymity parameter, one solution is: after receiving all votes, partition the n contests into some small number χ of subsets such that within each subset of contests, each voter has at least $k-1$ other voters who vote in exactly the same way in that subset of contests, for some parameter k. In the public bulletin board, the χ subsets of contests are released independently. Each voter needs to store only χ tracking numbers (one for each subset of contests), and k-anonymity is achieved.

The case for MISP corresponds to the scenario when a parameter k is given, and the goal is the minimize the number χ of subsets to achieve k-anonymity. For the dual problem MISPDual, the number χ of subsets is given, and the goal is to partition the contests into χ subsets such that the anonymity parameter k is maximized. Hence, it is of interest to investigate the complexity and hardness of approximation for these problems.

Our Results and Techniques. We prove strong inapproximability results for both problems MISP and MISPDual. We first give a reduction from graph coloring, which is NP-hard; in graph coloring, each vertex is assigned a color such that no two adjacent vertices receive the same color. Intuitively, each index in $[n]$ stands for a vertex, while the vectors capture the properties of the graph coloring problem. In our construction, a valid coloring corresponds to a partition with multiplicity k, while an invalid coloring corresponds to one with multiplicity 1.

The inapproximability of graph coloring implies that the approximation hardness of MISP and MISPDual with $\chi \geq 3$ is at least $n^{1-\epsilon}$ and $N^{1-\epsilon}$ respectively.

However, we show that MISPDual with $\chi = 2$ is much harder than graph coloring. Observe that deciding if a graph is 2-colorable can be solved in polynomial time. Hence, to show the hardness of MISPDual with $\chi = 2$, we need new reduction techniques. We give a novel reduction from the NP-hard problem 3-SAT. Similar to graph coloring, some indices stand variables and their negations. Intuitively, one subset stands for "true" and the other stands for "false". In order to show approximation hardness, for any threshold k, our reduction is carefully constructed such that a satisfiable assignment corresponds to a partition with multiplicity at least k, while an unsatisfiable formula corresponds to an instance such that any 2-partition has multiplicity only 1. This gap property allows us to prove that it is NP-hard to approximate MISPDual within factor $N^{1-\epsilon}$.

Our strong inapproximability results imply that there can be no efficient approximation algorithms for the problems MISP and MISPDual in their most general form. However, in real-world applications, the instances might have special structures that facilitate useful heuristic algorithms, which we leave as future research directions.

1.1 Historical Overview on Inapproximability

NP-Completeness has been developed in the 1970s [2, 9]. Its success motivated the study of approximation algorithms. The first such paper was by Johnson [8]. He considered the problems Max SAT, Independent Set, Coloring and Set Cover. Several approximation algorithms have been proposed for these problems in this paper.

The design and analysis of approximation algorithms have grown since then. Several problems are shown to admit polynomial time approximation schemes (PTAS), meaning that they can be approximated as close to the optimum as possible.

It was known from the very beginning of approximation algorithms that some problems do not admit PTAS. For instance, coloring can not be approximated within $\frac{4}{3} - \epsilon$, since 3-coloring is NP-hard. However, the inapproximabilities for many hard problems remains unknown.

Modern theory of inapproximability starts from the development of PCP systems, which are proved in [1]. Unlike conventional NP-hardness reduction, PCP systems can be used more readily to achieve inapproximability hardness. Based on PCP systems, several strong inapproximability have been proved since then, e.g., MAX 3SAT [7], Set Cover [5] and Coloring [4, 13]. In particular, one of our reduction is based on the hardness of graph coloring [13].

Other Vector Partition Problems. Onn and Schulman [10] have also considered vector partition problems in which the input is also a collection of vectors. However, the goal is to partition the vectors (as opposed the coordinate index set) to maximize some convex objective function on the sum of vectors in each part. They showed that if both the dimension and the number of parts are fixed, the problem can be solved in strongly polynomial time.

2 Problem Definition

We give the formal definition of the Minimum Independent Set Partition Problem (MISP).

The input is a positive integer n, a multi-set $Y := \{y_1, y_2, \ldots, y_N\}$ of N vectors in $\{0,1\}^n$, and a multiplicity threshold k. We use $U := [n] = \{1, 2, \ldots, n\}$ to denote the set of indices.

Given a vector y and subset $I \subseteq U$ of indices, we use $y|_X$ to denote the projection of vector y on I. For instance, for $y = (0,1,0,1,0,1)$ and $I = \{2,4,5\}$, $y|_I = (1,1,0)$.

Given a multi-set Y, the projection of Y on I is a multi-set defined similarly $Y|_I := \{y|_I : y \in Y\}$.

A subset $I \subseteq U$ of indices is *k-independent* (with respect to Y) if each vector in the multi-set $Y|_I$ has multiplicity at least k, where multiplicity denotes the number of times a vector in $Y|_I$ repeats. A partition $\{I_1, I_2, \ldots, I_\chi\}$ of U is k-independent if each part I_i is k-independent.

The goal is to find the smallest integer χ and partition U into χ subsets $I_1, \ldots, I_\chi$ such that each partition I_i is k-independent.

Dual Problem. We also describe a dual version of the problem that we call MISPDual. Similarly, a multi-set Y of vectors are given, and a target number χ of partitions is given instead of k.

The goal is to maximize k and partition the indices U into χ subsets $I_1, \ldots, I_\chi$ such that each I_i is k-independent.

3 General Reduction Schema

In this section, we reduce from the problem of graph coloring to MISP (and MISPDual with $\chi \geq 3$); in a *valid* coloring of an undirected graph, each vertex is assigned a color such that no two adjacent vertices receive the same color. We convert from an undirected graph $G = (V, E)$ to a multi-set of vectors such that a valid coloring corresponds to satisfying some fixed multiplicity threshold k while an invalid coloring leads to multiplicity 1. The use of this "k vs 1"-gap will be clear in the proof of the hardness of MISPDual. Because graph coloring is hard to approximate [13], our reduction readily implies the approximation hardness of MISP (and MISPDual with $\chi \geq 3$).

Our reduction depends on an arbitrarily chosen parameter $k > 1$ that is the same as the given threshold in MISP or may depend on the graph size $n = |V|$ in MISPDual. The index set is $U := [n]$. The multi-set Y consists of $N = k(n+1) + \binom{n}{2} + (k-1)|\overline{E}|$ vectors in $\{0,1\}^n$, where $\overline{E}$ is the edges in the complement graph of G. We also use $u \in V$ to denote an index of a vector. Let MISP(G, k) be the instance reduced from graph G with parameter k. The vectors in MISP(G, k) are defined as follows.

(I) An all-0's vector, and the n vectors in the standard basis (each having exactly one non-zero coordinate). Each of these vectors are repeated k times. There are $k(n+1)$ such vectors.

(II) Vectors of exactly two non-zero coordinates. There are $\binom{n}{2}$ such vectors.

(III) For each $(u, v) \notin E$, the vector with exactly two non-zero entries at indices u and v. Each of these vectors are repeated $(k-1)$ times. There are $(k-1)|\overline{E}|$ such vectors.

Figure 1 contains an example of the vectors for graph $G = (V = \{a, b, c, d\}, E = \{\{a, b\}, \{a, c\}, \{a, d\}\})$ and $k = 3$. Observe that parts (I) and (II) only depend on the size of graph G and k.

Fig. 1. An example of $G = (V = \{a, b, c, d\}, E = \{\{a, b\}, \{a, c\}, \{a, d\}\})$ and $k = 3$

	a	b	c	d
(I)	0	0	0	0
	0	0	0	0
	0	0	0	0
	1	0	0	0
	1	0	0	0
	1	0	0	0
	0	1	0	0
	0	1	0	0
	0	1	0	0
	0	0	1	0
	0	0	1	0
	0	0	1	0
	0	0	0	1
	0	0	0	1
	0	0	0	1
(II)	1	1	0	0
	1	0	1	0
	1	0	0	1
	0	1	1	0
	0	1	0	1
	0	0	1	1
(III). (b, c)	0	1	1	0
	0	1	1	0
(III). (b, d)	0	1	0	1
	0	1	0	1
(III). (c, d)	0	0	1	1
	0	0	1	1

Note that a coloring of the graph gives a partition on U (and vice versa) in a natural way, where vertices having the same color corresponds to a subset of indices. Next we prove the relationship between colorings and partitions.

Theorem 1. *For any $k > 1$ and graph G, G has a valid χ-coloring iff $\mathsf{MISP}(G, k)$ has a k-independent χ-partition. If G does not have any valid χ-coloring, then any χ-partition of $\mathsf{MISP}(G, k)$ is not 2-independent.*

Proof. When G has a valid χ-coloring, we can induce a χ-partition from the coloring. We prove it is k-independent.

Given a subset I of indices, consider the projected vectors in each part of the reduction.

Each projected vector in part (I) appears at least k times by the construction. Each projected vector in (III) appears at least k times since it repeats $k-1$ times in (III) and we can find a same one in (II).

For a vector in part (II), it depends on the indices u and v at which the entries are non-zero. If at most one of them is included in I, then the projected vector already appears k times in (I); otherwise, both u and v are included in I.

Two vertices u and v can be included in the same part I only if they are not neighbors in G; hence, the projected vector appears once from (II) and $k-1$ times from (III). This proves the "only if" part.

On the other hand, if a χ-partition is 2-independent, then we induce a χ-coloring for G from the partition. We claim the coloring is valid. For any vertices u, v with the same color, we have a vector in (II) with u, v in the same part I (the part corresponding to their color). Such vector appears only once in (II). It appears in (III) at least once, since the partition is 2-independent. Hence u, v cannot be neighbours in G, thus it is a valid coloring. Notice k-independent implies 2-independent. This proves the "if" part and the contrapositive proves the second statement.

Theorem 2. *The inapproximability of MISP is $n^{1-\epsilon}$ for arbitrarily small $\epsilon > 0$, unless $\mathsf{P} = \mathsf{NP}$; this means that if a k-independent partition has minimum number of parts χ, it is NP-hard to return a k-independent partition with at most $n^{1-\epsilon} \cdot \chi$ parts. Moreover, the result holds for any constant $k \geq 2$.*

Proof. We want to show a reduction from a coloring instance G to an instance of MISP.

Use the "reduction schema" in Theorem 1 with $k \geq 2$ to get a multi-set Y, which is an MISP instance with threshold k.

It is immediate from Theorem 1 that the minimum χ such that there is a k-independent partition with χ parts in the MISP instance is the same as the chromatic number of G (the minimum number of colors needed to color G).

Thus, the inapproximability of graph coloring can be applied to MISP. The inapproximability of chromatic number is $n^{1-\epsilon}$, by [13], meaning that it is NP-hard to approximate chromatic number within a factor of $n^{1-\epsilon}$. Hence, it is also NP-hard to approximate MISP within a factor of $n^{1-\epsilon}$.

4 Approximation Hardness of MISPDual

In this section, we show that the dual problem of maximizing the multiplicity of the projections into partitions with $\chi \geq 3$ is hard to approximate. In Section 5, we show that even for $\chi = 2$, the problem is hard.

Theorem 3. *For arbitrarily small constant $\epsilon > 0$, there is no polynomial time algorithm that approximates* MISPDual *within a factor of $N^{1-\epsilon}$, where N is the number of vectors in the given multi-set Y; moreover this result holds for any constant $\chi \geq 3$, unless* P=NP.

Remark 1. We comment on choosing "n vs N" as the parameter to express approximation hardness. In MISP, a trivial solution is to partition U into n singletons, and hence, it is natural to compare with the trivial solution with approximation ratio n. Hence, inapproximability within factor $n^{1-\epsilon}$ is a strong indicator that no useful algorithm would exist.

In MISPDual, since any partition would give multiplicity 1, and the maximum possible multiplicity is the number N of vectors, inapproximability within factor $N^{1-\epsilon}$ indicates that there is no useful algorithm. Observe that we can also derive n^C hardness for MISPDual for any constant C.

Proof. We use the fact [3] that the problem of deciding whether a graph is χ-colorable is NP-complete for any $\chi \geq 3$.

We reduce the problem of deciding whether a graph G is χ-colorable to MISPDual, such that for a "YES" instance, the multiplicity of MISPDual solution is at least k, otherwise the multiplicity is at most 1. Later we will set $k = n^C$ for some large enough constant $C = \Omega(\frac{1}{\epsilon})$.

Given a graph G, we use the "reduction schema" in Theorem 1 with $k = n^C$ to get a multi-set Y, which is an MISPDual instance with the same χ (target number of parts).

Suppose the graph is χ-colorable. From Theorem 1, we know that the MISPDual have solution with multiplicity at least k.

On the other hand, if the graph is not χ-colorable, then MISPDual only has solution with multiplicity 1, since otherwise it will contradict Theorem 1.

Note that the gap between "NO" and "YES" instances is 1 vs k.

We next prove no polynomial algorithm can approximate MISPDual within a factor better (smaller) than $k = n^C$. Note that the size N of Y is at most $k(n+1)+\binom{n}{2}+(k-1)|\overline{E}| \leq n^{C+10}$, hence this will imply no polynomial algorithm can approximate MISPDual within a factor better than $N^{\frac{C}{C+10}}$.

Suppose there is an algorithm $\mathcal{A}$ that can approximate MISPDual within a factor better than k. Then, we can decide whether a graph is χ-colorable by examining if the multiplicity is greater than 1. Hence, it is NP-hard to approximate MISPDual within a factor better than $k = n^C > N^{\frac{C}{C+10}}$. Note that for constant C, this is a polynomial-time reduction.

Setting C large enough such that $\frac{C}{C+10} > 1 - \epsilon$ gives the result.

5 Improved Approximation Hardness of MISPDual

This is the most technical part of the paper. In view of Section 4, it is natural to ask whether MISPDual with $\chi = 2$ is polynomial-time solvable, as deciding if a graph is 2-colorable has an easy solution.

In this section, we answer this question negatively. We show strong inapproximability result for MISPDual with $\chi = 2$. Observe that the reduction from graph coloring no longer works. To derive such a result, we need some problem with binary choice to tackle 2-partition. It turns out that 3-SAT does the job. In our construction the two parts correspond to "true" and "false" literals. At the same time, "true" and "false" literals are distinguishable via additional indices. The inapproximability comes from the fact that any satisfiable assignment corresponds to a 2-partition with high multiplicity, while any non-satisfiable assignment corresponds to a 2-partition with low multiplicity. In particular, we prove the following result.

Theorem 4. *For arbitrarily small constant $\epsilon > 0$, there is no polynomial algorithm that approximates* MISPDual *with $\chi = 2$ within a factor of $N^{1-\epsilon}$, unless* P=NP.

Proof. We use the fact that 3-SAT is NP-hard [9]. We construct a reduction from 3-SAT to MISPDual with $\chi = 2$. Consider an instance of 3-SAT: $C = \wedge_{i=1}^{l} C_i = \wedge_{i=1}^{l}(c_{i,1} \vee c_{i,2} \vee c_{i,3})$, with l clauses and p distinct variables.

Here $c_{i,j}$ can be x or $\neg x$. Without loss of generality, we assume that x and $\neg x$ do not appear in the same clause. It is obvious that $p \leq 3l$, and we further assume that $p, l \geq 2$ to avoid trivial cases.

The property of our reduction is that a satisfiable 3-SAT instance corresponds an MISPDual solution with multiplicity at least k (later fixed to be $l^{\Omega(\frac{1}{\epsilon})}$), while a non-satisfiable 3-SAT corresponds to an MISPDual solution with multiplicity at most 1. Notice that the gap "1 vs k" is used to derive the inapproximability result.

We next give the construction for the reduction from 3-SAT to MISPDual with $\chi = 2$. We need a parameter $k \geq 2$ to be fixed later, which will be polynomially related to l. We denote the resulting MISPDual instance by MISPDual(C, k), where C is the 3-SAT instance and k is the parameter.

Our reduction will generate a multi-set Y of vectors from $\{0,1\}^{(1+l+2p)}$, with index set $U := [l + 1 + 2p]$. The first l indices are identification indices and are denoted by $[1..l]$. The $(l+1)$-th index is the separation index and is denoted by $(l+1)$. The last $2p$ indices correspond to literals (and their negations) and are denoted by the literals, e.g., x or $\neg x$. The use of identification and separation indices will become clear in the proof.

NOTATION. To make the description easier, coordinates not mentioned are set to 0.

There are four parts of vectors as below:

(I) There are $2k$ vectors:
The 1st vector is the vector with the first l coordinates being 1.

The 2nd to the k-th vectors are the vectors with the first $l+1$ coordinates being 1.

The $(k+1)$-st vector is the vector with the $(l+1)$-st coordinate being 1.

The remaining $k-1$ vectors are all zero vectors.

The use of (I) is to force the identification indices $1..l$ to be in different part from the separation index $l+1$ in a "good" partition. Notice that some $(0,1)$ will appear only once otherwise.

(II) There are $(2k+1)p$ vectors. For each variable x, we have $(2k+1)$ vectors described as below:

(II.x) The first k vectors are the vectors with coordinates $(x, \neg x)$ being $(0,1)$.
The next k vectors are the vectors with coordinates $(x, \neg x)$ being $(1,0)$.
The last vector is a vector with indices $(x, \neg x)$ being $(1,1)$.

The use of (II) is to force x and $\neg x$ to be apart. Since there will be only one $(1,1)$ if the two indices are put together. In a "good" partition, literals setting to be "true" are supposed to be within the identification indices' (the first l indices) partition, while the "false" are in the separation index's (the $l+1$-st index) partition.

(III) There are $(3k+1)l$ vectors. For each clause $C_i = x \vee y \vee z$ (with literals x, y and z), we have $3k+1$ vectors:

(III.i) The first k vectors are the vectors with the i-th coordinate set to 1 and coordinates $(\neg y, \neg z)$ set to 1.
The next k vectors are the vectors with the i-th coordinate set to 1 and the coordinates $(\neg x, \neg z)$ set to 1.
The next k vectors are the vectors with the i-th coordinate set to 1 and the coordinates $(\neg x, \neg y)$ set to 1.
The last vector is the vector with the i-th coordinate set to 1 and the coordinates $(\neg x, \neg y, \neg z)$ set to 1.
Note that for all the $(3k+1)$ vectors, the i-th coordinate is set to 1.

The use of (III) is to force the variables to satisfy the constraints. Notice that if a clause is not satisfied, then all the indices $\neg x, \neg y, \neg z$ are on the "true" side (together with the first l indices), causing $(1,1,1)$ to appear only once in the projection onto the coordinates $(\neg x, \neg y, \neg z)$. On the other hand, as long as the not all indices $\neg x, \neg y, \neg z$ are included on the "true" side, any vector will appear at least k times. Notice the use of identification indices (the first l indices) here. With different identification indices, clauses will not affect each other.

(IV) There are lk vectors. For each clause $C_i = x \vee y \vee z$ there are k vectors as follows.

(IV.i) The k vectors are the same with the coordinates $(\neg x, \neg y, \neg z)$ set to 1.

Notice that in (IV) the identifier columns are set to 0, which is different from (III). The idea is to handle the situation when in (III.i) all $\neg x, \neg y, \neg z$ are partitioned into the "false" side. If this happens, the vector (projected on the "false" side with the $(l+1)$-st index) will repeat at least k times.

Figure 2 (in appendix) gives an example for $(x \vee y \vee z) \wedge (\neg y \vee \neg x \vee w)$ with $k = 2$.

It remains to show that a satisfiable assignment corresponds to an k-independent partition, while a non-satisfiable assignment corresponds to an MISPDual instance such that any 2-partition is not 2-independent.

Lemma 1. *For all $k > 1$ and 3-SAT instance C, if C has a satisfiable assignment, then* MISPDual(C, k) *has a k-independent 2-partition.*

Proof. Give a satisfiable assignment, we partition the indices set U into 2 subsets T and F as follows. The first l indices $[1..l]$ are included in T, and the $(l+1)$-st index is in F. For each literal x, if $x = true$ then the index x is included in T and the index $\neg x$ is included in F; otherwise, the index $\neg x$ is in T and the index x is in F.

We next consider the vectors in each part projected on T and F.

Claim. In (I), each vector appears at least k times on both T and F.

Proof. First we consider the each vector in (I) projected on T. By construction, in the first l coordinates, each of the all 1's and all 0's vectors is repeated k times, and other coordinates are all set to 0.

For the projections on F, only the $(l+1)$-st index has non-zero values and it contains exactly k 1's and k 0's. Hence, in (I), each projected vector repeats at least k times.

Claim. In (II.x), each vector appears at least k times on both T and F.

Proof. It can be seen that the only non-zero values are at indices x and $\neg x$. At both x and $\neg x$, we have more than k 0's and k 1's.

By the construction we know that x and $\neg x$ are assigned to different parts. In each part, the only non-zero coordinate is repeated at least k times, for each of the two values 0 and 1.

Claim. In (III.i), each vector appears at least k times on both T and F.

Proof. We denote the i-th clause by $C_i = x \vee y \vee z$, where x, y, z can be a variable or its negation. By construction, at least 1 of $\neg x, \neg y, \neg z$ is in F, since it is a satisfiable assignment. For instance, suppose $\neg z$ is in F; other situations follow the same argument.

Consider projections on F. Since the first l indices are not in F, we can find at least k same vectors in (III.i) and (IV.i) (in case $\neg x, \neg y, \neg z \in F$).

Now consider the projections on T. Vectors in (III.i) projected on T only differ at indices $\neg x$ and $\neg y$. It can be seen from the construction that no matter which part each of the indices $\neg x$ and $\neg y$ goes, each projected vector still appears at least k times.

Hence, result of Claim 5 follows.

Claim. In (IV), each vector appears at least k times on both T and F.

Proof. This follows immediately from the construction.

The result of Lemma 1 follows, since each projected vector repeats at least k times.

Lemma 2. *For all $k > 1$ and 3-SAT instance C, if $\mathsf{MISPDual}(C, k)$ has a 2-independent 2-partition, then C has a satisfiable assignment.*

Proof. We first argue that if the 2-partition is 2-independent, then the identification (first l) indices and the separation ($l+1$-st) index should be in different subsets. Similarly, x and $\neg x$ should be in different subsets. Then, an assignment is derived (such that literals on the same side as the identification indices are set to true) and analyzed.

Claim. Each of the indices $[1..l]$ is in the subset different from the subset containing the index $l+1$.

Proof. Note that the only 1's at index $l+1$ happens in vectors 2 to $k+1$. Suppose on the contrary that some index in $j \in [1..l]$ is in the same subset at index $l+1$. Then, at the coordinates $(j, l+1)$, the projection $(0, 1)$ will appear only once due to vector $k+1$. This contradicts 2-independence.

We denote T as the subset containing $[1..l]$, and F as the other subset F.

Claim. For each literal x, x and $\neg x$ are in different subsets.

Proof. Notice that we assume that no x and $\neg x$ appear in the same clause. As a result, there will be no vector with coordinates $(x, \neg x)$ being $(1, 1)$ in (I,III,IV). Such a vector appears only once in (II). The result follows as the partition is 2-independent.

From this point it is obvious that we should assign $true$ to the literals in T and $false$ to the literals in F. Next we prove that this is indeed a satisfying assignment.

Claim. Every clause C_i is satisfied by the above assignment.

Proof. Suppose $C_i = x \vee y \vee z$ is not satisfied. Then, it must be the case that $\neg x, \neg y, \neg z \in T$. We consider the vectors in (III.i) projected on T. From the construction, in (III.i) there will be exactly one vector with coordinates $(\neg x, \neg y, \neg z)$ being $(1, 1, 1)$.

We argue that this vector projected on T does not appear anywhere else. To see this, note that the identification indices are included in T, which is different from all other parts except (III.i). In (III.i), such vector (projected on T) only appears once, and hence the result follows.

This completes the proof of Lemma 2.

The following corollary is the contrapositive of Lemma 2.

Corollary 1. *For all $k > 1$ and 3-SAT instance C, if C does not have any satisfiable assignment, then any 2-partition for $\mathsf{MISPDual}(C, k)$ is not 2-independent.*

At this point, we can see that there is a gap of 1 vs k, meaning that to distinguish satisfiable 3-SAT from unsatisfiable ones, we only need to distinguish between multiplicity 1 and k. Hence, any polynomial algorithm that approximates MISPDual within a factor better than k will imply P=NP.

We can set $k = l^C$ for some large enough constant C, and observing that $N \leq l^{C+10}$, we conclude that there is no polynomial algorithm with approximation ratio better than $N^{\frac{C}{C+10}}$.

Choosing C large enough (depending on ϵ) completes the proof of Theorem 4.

Acknowledgment. We would like to thank David Wagner for posting the problem online [12] and for useful discussions.

References

1. Sanjeev Arora and Shmuel Safra. Probabilistic checking of proofs: A new characterization of np. *J. ACM*, 45(1):70–122, January 1998.
2. Stephen A. Cook. The complexity of theorem-proving procedures. In *Proceedings of the Third Annual ACM Symposium on Theory of Computing*, STOC '71, pages 151–158, New York, NY, USA, 1971. ACM.
3. David P. Dailey. Uniqueness of colorability and colorability of planar 4-regular graphs are np-complete. *Discrete Mathematics*, 30(3):289 – 293, 1980.
4. U. Feige and Joe Kilian. Zero knowledge and the chromatic number. In *Computational Complexity, 1996. Proceedings., Eleventh Annual IEEE Conference on*, pages 278–287, 1996.
5. Uriel Feige. A threshold of ln n for approximating set cover. *J. ACM*, 45(4):634–652, July 1998.
6. Srivatsava Ranjit Ganta, Shiva Prasad Kasiviswanathan, and Adam Smith. Composition attacks and auxiliary information in data privacy. In *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '08, pages 265–273, New York, NY, USA, 2008. ACM.
7. Johan Håstad. Some optimal inapproximability results. *J. ACM*, 48(4):798–859, July 2001.
8. David S. Johnson. Approximation algorithms for combinatorial problems. In *Proceedings of the Fifth Annual ACM Symposium on Theory of Computing*, STOC '73, pages 38–49, New York, NY, USA, 1973. ACM.
9. R. M. Karp. Reducibility Among Combinatorial Problems. In R. E. Miller and J. W. Thatcher, editors, *Complexity of Computer Computations*, pages 85–103. Plenum Press, 1972.
10. Shmuel Onn and Leonard J. Schulman. The vector partition problem for convex objective functions. *Math. Oper. Res.*, 26(3):583–590, 2001.
11. Latanya Sweeney. k-anonymity: A model for protecting privacy. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 10(5):557–570, 2002.
12. David Wagner. Find index set partition that has large projections. http://cstheory.stackexchange.com/questions/17562/find-index-set-partition-that-has-large-projections, 2013.
13. David Zuckerman. Linear degree extractors and the inapproximability of max clique and chromatic number. *Theory of Computing*, 3(6):103–128, 2007.

Appendix

Fig. 2. An example for $(x \vee y \vee z) \wedge (\neg y \vee \neg x \vee w)$ with $k = 2$; unspecified entries are 0.

	Identifiers		Separator	x	¬ x	y	¬ y	z	¬ z	w	¬ w
(I)	1	1	0								
	1	1	1								
	0	0	1								
	0	0	0								
(II.x)				0	1						
				0	1						
				1	0						
				1	0						
				1	1						
(II.y)						0	1				
						0	1				
						1	0				
						1	0				
						1	1				
(II.z)								0	1		
								0	1		
								1	0		
								1	0		
								1	1		
(II.w)										0	1
										0	1
										1	0
										1	0
										1	1
(III.1)	1	0			0		1		1		
	1	0			0		1		1		
	1	0			1		0		1		
	1	0			1		0		1		
	1	0			1		1		0		
	1	0			1		1		0		
	1	0			1		1		1		
(III.2)	0	1		1		0					1
	0	1		1		0					1
	0	1		0		1					1
	0	1		0		1					1
	0	1		1		1					0
	0	1		1		1					0
	0	1		1		1					1
(IV.1)	0	0			1		1		1		
	0	0			1		1		1		
(IV.2)	0	0		1		1					1
	0	0		1		1					1